“Instruction-Tuned LLMs for Construction and Utility Information Systems”

*DS 6051 Final Project - April 2025

https://github.com/kianadane/divulge

A Preprint

Kiana Dane
University of Virginia
urnh8e@virginia.edu

May 7, 2025

1 Abstract

Early-phase construction projects in dense urban environments are frequently delayed by outdated and disconnected subterranean infrastructure records, leading to accidental utility strikes and costly sitework revisions. This project addresses the problem within the scope of the City of San Francisco by developing a fine-tuned tool-calling language model for subsurface data retrieval. A Hugging Face pretrained base model is fine-tuned on thousands of synthetic instruction-response examples, automatically mined and filtered using a large language model seeded with domain-specific keywords related to excavation, permitting, utilities, and site preparation. The system is designed to parse natural language queries, invoke structured tools, and retrieve relevant permit and infrastructure metadata with minimal hallucination. By tightly integrating tool-calling capabilities during fine-tuning, the model can dynamically route queries to validated permit and site datasets. The resulting agent aims to support early construction planning by providing scalable, accurate, and queryable access to fragmented subterranean information, reducing the risk of unexpected underground conflicts and expediting the permitting process.

2 Introduction

Delays and unexpected costs in the early phases of construction projects are frequently caused by inaccurate or fragmented subsurface utility data. Struck utility lines, unanticipated infrastructure conflicts, and incomplete permit histories remain persistent challenges that can derail site preparation and excavation activities. These issues not only inflate project timelines and budgets but also pose safety risks to construction crews and surrounding communities. Addressing these early-phase vulnerabilities requires more intelligent, context-aware access to existing site and permitting information. This project focuses specifically on new construction projects within the City and County of San Francisco, an urban environment where underground utility complexity and historical infrastructure records present unique challenges. The objective is to develop a fine-tuned language model that can intelligently call external tools to query ingested permit numbers, site survey data, and other relevant records, enabling faster and more reliable early-stage planning. The project approach leverages recent advancements in instruction tuning and tool-calling architectures. Using open-access language models from Hugging Face as a foundation, a large, domain-specific synthetic dataset was generated through targeted large language model (LLM) prompting. This dataset emphasizes excavation, utilities, permits, site surveys, and related early construction concerns. Fine-tuning the base model on this curated dataset produces a tool-calling capable model optimized for intelligently assisting with subsurface and permitting queries in a geographically constrained context.

3 Related Works

This project builds upon recent advancements in synthetic instruction tuning, lightweight fine-tuning of open-source language models, tool-augmented LLM capabilities, and urban subsurface infrastructure management. For data generation, self-instruct methodologies Y. Wang et al. 2022 and the Alpaca framework Taori et al. 2023 demonstrated that synthetic instruction-response pairs can significantly improve the alignment of language models. This research adopts a similar strategy by generating construction- and utilities-focused examples using keyword-driven LLM prompting, with additional mining from the RedPajama dataset Computer 2023 to ensure domain relevance. Fine-tuning small, open models has proven effective for domain adaptation, as evidenced by Alpaca /citetaori2023stanford and Vicuna Chiang et al. 2023. Following this line of work, a fine-tuned lightweight model (Mistral-7B-Instruct-v0.3) on a specialized instruction set, balancing computational efficiency with performance in specialized tool-calling tasks. Recent studies on tool-augmented language models, including Toolformer Schick et al. 2023, show that LLMs can be trained to invoke external tools via structured outputs. Similar principles are applied here, with the model being fine-tuned to produce tool-calling JSON structures that interact with permit and site anomaly retrieval APIs. The problem context draws from challenges in subsurface utility engineering, where outdated and incomplete underground data often leads to costly accidents during construction Sterling 2009. Addressing this issue, our system aims to improve early-stage site intelligence by enabling automated, accurate access to relevant permitting and site survey information.

See Section 3.

4 Methodology

The specific goals of the project included:

- **Fine-Tune a Foundation LLM on Domain-Specific Data**
  - Leverage a strong open-source instruction-tuned model (Mistral-7B-Instruct) and apply parameter-efficient fine-tuning (PEFT) via LoRA to adapt it for a new task: generating natural language hypotheses from sensor readings.
- **Build a Synthetic Supervised Dataset**
  - Create a dataset of paired examples where each input is a concise sensor observation (e.g., chemical measurement, location, date) and the output is a hypothesis that plausibly links the anomaly to nearby construction activity, inferred from permit metadata.
- **Optimize Training Pipeline for Low-Resource Environments**
  - Use 4-bit quantization and gradient accumulation to fine-tune the model on a single GPU with limited memory (e.g., <15GB), making the workflow accessible and replicable on modest cloud infrastructure.
- **Evaluate Model Outputs Using Automatic Metrics and Human Review**
  - Assess the model's ability to generate high-quality hypotheses using both automatic NLP metrics (BERTScore, ROUGE, BLEU) and qualitative inspection of outputs for coherence, relevance, and plausibility.

4.1 Dataset Construction

To train the model to generate plausible causal hypotheses linking environmental sensor anomalies to construction activity, a synthetic supervised dataset was constructed from two distinct sources: (1) environmental sensor measurements and (2) construction permit records.

Since no existing dataset contained paired examples of sensor readings and natural language hypotheses, a pipeline was developed to automatically generate and curate these training examples.

4.1.1 Source Data

Environmental Sensor Data: A dataset of time-stamped environmental measurements collected at specific street-level locations. Each record included:

- Date of measurement
- Location (typically a street or address)
- Characteristic name (e.g., Ammonia-nitrogen, Nitrate, Total Suspended Solids)

    - Measurement value and units

Construction Permit Data: Construction permits issued across the same geographic area and time span were also ingested via API and processed. Each permit record included:

    - Permit issue date
    - Permit type (e.g., sewer installation, demolition, grading)
    - Street address or location of activity
    - Optional description fields

4.2 Preprocessing and Normalization

The sensor and permit data were cleaned, filtered, and normalized as follows:

- Sensor measurements with missing or invalid values were removed.
- Location names were standardized to street-level identifiers to enable cross-dataset joins.
- Permit descriptions were parsed for construction-related keywords (e.g., "sewer," "excavation") and filtered to focus on activities likely to affect environmental quality.

4.3 Synthetic Example Generation

Synthetic input-output pairs were generated by joining sensor readings with nearby construction permits. A pairing was considered valid if:

- The sensor measurement occurred within 7 days of the permit's `ActivityStartDate`.
- The location matched or was closely related (e.g., the same street name).

Each example was structured as:

- **Input:** A sentence summarizing the sensor measurement.
  Example: "Detected Nitrate at 0.27 mg/L on WESTGATE DR during 2000-07-31."
- **Output:** A question-style hypothesis linking the measurement to nearby construction.
  Example: "Could the increased nitrate levels be related to the sewer line excavation that was conducted during the same period?"

Over 600 cleaned examples were generated. A sample was manually reviewed to assess grammaticality, coherence, and factual consistency.

4.4 Formatting for Instruction Tuning

The dataset was converted to a ChatML-style structure for compatibility with instruction-tuned language models. Each example was formatted as:

```
{
  "messages": [
    {"role": "user", "content": "<sensor report>"},
    {"role": "assistant", "content": "<hypothesis question>"}
  ]
}
```

This format aligned with the instruction-following behavior of the Mistral-Instruct model and was used directly during training.

4.5 Model Training

To train the model to generate plausible causal hypotheses grounded in sensor data, we fine-tuned a pretrained instruction-tuned language model using our synthetic dataset.

`Mistral-7B-Instruct-v0.3`, a 7-billion parameter instruction-tuned model, was chosen due to its strong performance on reasoning and QA tasks, as well as its compatibility with parameter-efficient tuning methods. LoRA

(Low-Rank Adaptation) was applied for parameter-efficient fine-tuning. LoRA introduces trainable low-rank matrices into selected attention layers, reducing GPU memory usage and training time.

- LoRA rank: 8
- LoRA alpha: 16
- LoRA dropout: 0.1
- Task type: Causal Language Modeling (`CAUSAL_LM`)

The base model was loaded in 4-bit precision using Hugging Face's `BitsAndBytesConfig`, allowing training on a single 15–16 GB GPU with partial CPU offloading.

4.5.1 Training Setup

`SFTTrainer` from the `trl` library (a wrapper around `transformers.Trainer`) was used to perform supervised fine-tuning with ChatML-formatted data.

Training configuration:

- Epochs: 10
- Batch size: 4 (per device)
- Gradient accumulation steps: 2
- Learning rate: `2e-4`
- Mixed precision: FP16
- Checkpoints: Saved at the end of each epoch

Training was performed on an AWS g4dn.xlarge instance, which includes a single NVIDIA T4 GPU (16 GB). The total training process took approximately 15 minutes across 10 epochs and 60 examples in the held-out evaluation set.

5 Experiments and Results

5.1 Evaluation Setup

After training, the performance of the fine-tuned model was evaluated using a held-out test set comprising 10% of the synthetic dataset. For each test example, the model generated a hypothesis from a sensor report prompt, following the same ChatML-style formatting used during training. The predictions were compared against reference outputs using standard natural language generation (NLG) metrics.

5.2 Quantitative Metrics

Three widely accepted metrics were used to assess the quality of generated hypotheses:

- **BERTScore (F1):** 0.9377
  Captures semantic similarity between model outputs and references using contextual embeddings. A high score indicates strong semantic alignment, even if surface phrasing differs.
- **ROUGE-L (F1):** 0.5861
  Measures lexical overlap based on longest common subsequences. Reflects fluency and content consistency, though it is sensitive to word order and phrasing.
- **BLEU:** 0.3724
  Computes n-gram overlap with penalties for word order deviations. Useful for measuring surface-level similarity and stylistic fidelity.

The high BERTScore suggests that the model captures the intended meaning of the target questions, while moderate ROUGE and BLEU scores indicate variability in phrasing rather than failure to model the task.

5.3 Qualitative Analysis

A manual review of a sample of generated hypotheses revealed that the model generally performs well in:

- Referencing the correct **sensor type** and measured **value**.
- Incorporating **time** and **location** details from the input.
- Suggesting plausible **construction-related causes** such as sewer excavation or concrete pouring.

Example Output:

> *Prompt:* Detected Total suspended solids at 6.0 mg/L on WESTGATE DR during 2000-07-17.
> *Model Output:* Could the increased total suspended solids at 6.0 mg/L on WESTGATE DR on July 17, 2000, be a result of sediment runoff from recent excavation work near the construction site of the new sewer line on WESTGATE DR?

This example demonstrates factual grounding, linguistic fluency, and causal reasoning aligned with the task objective. The model exhibits generalization rather than memorization, as evidenced by varied yet accurate phrasings across samples.

5.4 Failure Modes

A small number of outputs exhibited the following issues:

- **Redundancy:** Repetition of phrases (e.g., restating location or date).
- **Speculation:** Vague or unfounded causal links without construction context.
- **Truncation:** Incomplete generations due to token limit constraints.

Future work may address these issues by:

- Applying length-controlled decoding strategies.
- Introducing post-processing filters to eliminate redundancy.
- Exploring reinforcement learning with human feedback (RLHF) to fine-tune causal inference behavior.

6 Discussion

The experimental results highlight the importance of high-quality synthetic data and domain-specific instruction tuning in shaping structured LLM behavior. The curated dataset, created through targeted prompting and filtering on construction- and utilities-related topics, enabled the model to learn robust tool usage patterns. Even when faced with complex or loosely phrased prompts, the model consistently produced valid JSON calls for relevant tools, suggesting strong generalization within the trained domain.

Notably, the model demonstrated the ability to disambiguate between similar but distinct tasks, such as distinguishing a *permit status check* from a *subsurface survey report request*, even when both were referenced in the same input. This suggests that the fine-tuning process not only imparted tool syntax but also instilled basic task reasoning aligned with domain expectations.

Limitations

Several limitations were observed:

- **Geographic bias:** The training assumed knowledge of San Francisco-specific permitting processes and utilities. Application to other regions would likely require additional fine-tuning on location-specific data.
- **Synthetic data imperfections:** Despite extensive filtering, occasional artifacts from the synthetic generation process remained, leading to minor but noticeable inaccuracies in rare edge cases.
- **Limited multi-modality:** The current approach is purely text-based. Many real-world subsurface investigations involve spatial or visual data (e.g., geophysical scans), which our system cannot yet interpret.

Future Work

Future directions to enhance the system include:

- **Live API integration:** Connecting to active municipal permit databases or inspection scheduling systems would enable real-time dynamic reasoning and decision-making.
- **Adversarial robustness:** Expanding the dataset with adversarially generated prompts could improve the model's resilience to ambiguous or under-specified queries.
- **Multimodal extension:** Incorporating geospatial layers, subsurface imaging (e.g., ground-penetrating radar, LiDAR), and site blueprints would allow the agent to achieve richer, cross-modal situational awareness and inference.

Overall, these findings demonstrate that targeted fine-tuning with structured synthetic data is a highly effective strategy for adapting open-source LLMs to specialized, tool-driven industrial domains like construction and utilities.

7 Bibliography

References

Chiang, Lulu et al. (2023). *Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90%* ChatGPT Quality*. `https://lmsys.org/blog/2023-03-30-vicuna/`.

Computer, Together (2023). *RedPajama: Open pretraining data for large language models*. `https://github.com/togethercomputer/RedPajama-Data`.

Schick, Timo et al. (2023). "Toolformer: Language Models Can Teach Themselves to Use Tools". In: *arXiv preprint arXiv:2302.04761*.

Sterling, Raymond L. (2009). "Underground infrastructure research: Strategic directions". In: *Proceedings of the ASCE International Conference on Pipelines*, pp. 1–10.

Taori, Rohan et al. (2023). *Stanford Alpaca: An Instruction-following LLaMA Model*. `https://github.com/tatsu-lab/stanford_alpaca`.

Wang, Yizhong et al. (2022). "Self-Instruct: Aligning Language Models with Self-Generated Instructions". In: *arXiv preprint arXiv:2212.10560*.